



FINAL PROJECT - TE 141599

DETEKSI *SALIENCY* PADA PERANGKAT BERGERAK

Arif Khumaidi
NRP 2207100702

Dosen Pembimbing
Dr. Surya Sumpeno, ST., M.Sc.
Reza Fuad Rachmadi, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

SALIENCY DETECTION ON MOBILE DEVICE

Arif Khumaidi
NRP 2207100702

Counselor Lecturer:
Dr. Surya Sumpeno, ST., M.Sc.
Reza Fuad Rachmadi, ST., MT.

ELECTRICAL ENGINEERING
Faculty of Industrial Engineering
Institut Teknologi Sepuluh Nopember
Surabaya 2015

DETEKSI SALIENCY PADA PERANGKAT BERGERAK

TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Teknik Komputer dan Telematika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I



Dosen Pembimbing II



Dr. Surya Sumpeno, ST., M.Sc.

NIP. 196906131997021003

Reza Fuad Rachmadi, ST., MT.

NIP. 198504032012121001



**SURABAYA
JANUARI, 2015**

ABSTRAK

Saliency didefinisikan sebagai sebuah informasi pada bagian tertentu lebih diperhatikan, bermakna dan berkesan dibandingkan dengan bagian-bagian tetangganya. Deteksi *Saliency* dianggap sebagai kunci dari mekanisme perhatian penglihatan manusia, karena memungkinkan untuk memfokuskan sumber daya persepsi dan kognitif mereka yang terbatas. Kemampuan sistem visual manusia untuk mendeteksi visual *saliency* sangat cepat dan handal, teknologi komputer diharapkan untuk bisa mendekati sistem visual *saliency* pada manusia. Untuk itu digunakanlah metode deteksi *saliency* berbasis model *Spectral residual* dari Hou dan Zhang. Metode ini memerlukan waktu kurang dari satu detik untuk bisa mendeteksi objek yang dianggap *saliency* dengan nilai akurasi sebesar 84,1%, namun sebagian besar nilai akurasi berasal dari kesuksesan model dalam mendeteksi objek yang bukan *saliency*. Dalam melakukan percobaan deteksi *saliency*, seseorang harus melewati beberapa kesulitan, diantaranya adalah membeli seperangkat komputer dan mencari dataset, sehingga harus meluangkan waktu dan menghabiskan biaya yang tidak sedikit. Untuk mengatasi masalah tersebut maka dibuatlah aplikasi *mobile* berplatform android agar pemodelan komputasi untuk deteksi *saliency* bisa digunakan kapan dan dimana saja dengan dataset yang tidak terbatas.

Kata kunci : *Saliency*, Android, *Spectral residual*

ABSTRACT

Saliency is the state or quality of an object which it stands out relative to its neighbors. Saliency detection is considered to be a key attentional mechanism that facilitates learning and survival by enabling organisms to focus their limited perceptual and cognitive resources. The ability of human visual system to detect visual saliency is extraordinarily fast and reliable, computer technology is expected to be able to approach visual saliency in human visual system. As for that, saliency detection method is used based on the residual spectral models by Hou and Zhang. This method takes less than one second to be able to detect objects saliency with 84.1% accuracy rate but the value of accuracy is largely derived from the success of the model in detecting objects that are not saliency. To perform saliency detection experiments, one must pass through some difficulties, such as buying a computer and search for datasets, so it should take the time and the cost is not small. To solve the problem then be made android mobile application platform for saliency detection that can be used anytime and anywhere with unlimited datasets.

Keyword : *Saliency, Android, Spectral residual*

KATA PENGANTAR

Puji dan Syukur penulis panjatkan kepada Allah SWT atas segala limpahan berkah, rahmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian ini. Penelitian ini disusun untuk memperoleh gelar Sarjana dalam Bidang Studi Komputer dan Telematika, Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua Orang Tua, yang telah memberikan dorongan secara material maupun spiritual sehingga dapat terselesaikannya penelitian ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember
3. Bapak Dr. Surya Sumpeno, ST., M.Sc. dan Bapak Reza Fuad Rachmadi, ST., MT., selaku dosen pembimbing dalam penelitian ini, yang telah memberikan bimbingan dan saran hingga terselesaikannya penelitian ini.
4. Laily Maziyah, yang telah membantu dan memberikan dorongan hingga dapat terselesaikannya penelitian ini.
5. Syakirul Adhim dan Lutfi Muhlasin yang dengan kesabarannya membantu terselesaikannya penelitian ini.
6. Bapak-Ibu dosen pengajar Bidang Studi Teknik Komputer dan Telematika, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
7. Seluruh teman-teman angkatan e-47, serta teman-teman B201crew Laboratorium Bidang Studi Teknik Komputer dan Telematika.

Penulis menyadari dalam penyusunan buku penelitian ini masih banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan buku penelitian ini. Semoga penelitian ini dapat memberikan manfaat bagi kita semua.

Surabaya, Januari 2015

Penulis

DAFTAR ISI

ABSTRAK	I
ABSTRACT	III
KATA PENGANTAR.....	V
DAFTAR ISI.....	VII
DAFTAR GAMBAR.....	IX
DAFTAR TABEL	XI
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Permasalahan	1
1.3. Tujuan dan Manfaat	2
1.4. Batasan Masalah	2
1.5. Sistematika Penulisan.....	2
BAB 2 TEORI PENUNJANG.....	5
2.1. <i>Saliency</i>	5
2.2. Log Spektrum	6
2.3. <i>Spectral residual</i>	8
2.4. <i>Saliency Map</i>	9
2.5. Transformasi Fourier	10
2.5.1. Transformasi Fourier kontinu	11
2.5.2. Transformasi Fourier Diskrit	11
2.6. Android	12
2.6.1. OpenCV.....	12
2.6.2. JNI.....	13
2.7. <i>Smoothing</i>	14
2.8. <i>Precision dan Recall</i>	14
BAB 3 DESAIN SISTEM DAN IMPLEMENTASI	17
3.1. Desain Sistem	17
3.1.1. Alur Kerja Sistem	17

3.1.2.	Prinsip Kerja Sistem.....	18
3.2.	Implementasi Sistem	20
3.2.1.	Penyusunan Program.....	20
3.2.2.	Deteksi <i>Saliency</i> Pada Android.....	22
3.2.3.	Perhitungan evaluasi model deteksi	23
3.3.	Desain Sistem Survei	24
BAB 4 PENGUJIAN DAN ANALISA.....		29
4.1.	Pengujian Aplikasi	29
4.1.1.	Pengujian <i>Saliency</i> Pada Android	30
4.1.2.	Pengujian <i>Saliency</i> Pada Komputer	33
4.2.	Analisa Hasil Survei.....	36
BAB 5 PENUTUP		41
5.1.	Kesimpulan.....	41
5.2.	Saran	41
DAFTAR PUSTAKA		43
LAMPIRAN.....		45
BIOGRAFI PENULIS		59

DAFTAR TABEL

TABEL 4-1 SPESIFIKASI PERANGKAT UJI ANDROID LENOVO A369i	29
TABEL 4-2 SPESIFIKASI PERANGKAT UJI ANDROID ZENFONE 4	29
TABEL 4-3 SPESIFIKASI PERANGKAT UJI KOMPUTER.....	30
TABEL 4-4 HASIL PERBANDINGAN WAKTU DETEKSI <i>SALIENCY</i> PERANGKAT	32
TABEL 4-5 PERFORMANSI HASIL UJI MODEL PADA DATASET	33
TABEL 4-6 PERBANDINGAN HASIL <i>SALIENCY</i> PADA ANDROID DAN KOMPUTER	35
TABEL 4-7 HASIL SURVEI MASYARAKAT TERHADAP CITRA	36
TABEL 4-8 RANGKUMAN HASIL SURVEI TERHADAP APLIKASI	39

DAFTAR GAMBAR

GAMBAR 2.1 CONTOH OBJEK <i>SALIENCY</i> . [1].....	6
GAMBAR 2.2 KURVA LOG SPEKTRUM.....	7
GAMBAR 2.3 KURVA SPEKTRUM 1, 10, DAN 100 CITRA	7
GAMBAR 2.4 PROSES MENDAPATKAN <i>SALIENCY MAP</i> DARI SUATU CITRA	9
GAMBAR 2.5 DIAGRAM ALIR <i>SALIENCY MAP</i>	10
GAMBAR 2.6 CONTOH APLIKASI OPENCV PADA ANDROID.....	13
GAMBAR 2.7 PRINSIP KERJA JAVA DAN NATIVE C DI ANDROID	13
GAMBAR 3.1 ALUR KERJA SISTEM SECARA UMUM	17
GAMBAR 3.2 TAMPILAN APLIKASI KETIKA INGIN MENGAMBIL SUATU CITRA.....	18
GAMBAR 3.3 DIAGRAM ALUR PRINSIP KERJA SISTEM	19
GAMBAR 3.4 VISUALISASI LOG SPEKTRUM SEBUAH CITRA.....	20
GAMBAR 3.5 <i>SPECTRAL RESIDUAL</i>	21
GAMBAR 3.6 HASIL DETEKSI OBJEK <i>SALIENCY</i> PADA PERANGKAT	22
GAMBAR 3.7 HASIL DETEKSI <i>SALIENCY</i>	22
GAMBAR 3.8 TEKNIK UNTUK MENDAPATKAN CITRA <i>TP</i> (ATAS) DAN <i>FP</i> (BAWAH). ..	23
GAMBAR 3.9 CONTOH FORM SURVEI CITRA PERTAMA	25
GAMBAR 4.1 HASIL DETEKSI <i>SALIENCY</i> PADA ANDROID	31
GAMBAR 4.2 DDMS PROFILING TOOL PADA ECLIPSE	31
GAMBAR 4.3 CONTOH HASIL DETEKSI <i>SALIENCY</i> PADA KOMPUTER	34

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Suatu citra umumnya memiliki bagian tertentu yang dianggap lebih menarik dan bermakna dibandingkan bagian lain dari citra tersebut. Inilah yang disebut dengan *saliency*. Kemampuan penglihatan manusia dalam mendeteksi posisi *saliency* sangat cepat dan handal. Hal ini dikarenakan manusia memiliki kecepatan yang tinggi saat mengolah sinyal yang diterima oleh mata, yang diteruskan ke dalam otak, untuk diproses dan diinterpretasi. Proses yang sangat rumit terjadi dengan begitu cepat dan belum tertandingi oleh mesin apapun saat ini.

Teknologi komputer visi diharapkan bisa membuat pemodelan dari sistem penglihatan manusia tersebut. Salah satu aplikasi yang dapat diterapkan dari deteksi *saliency* suatu citra adalah melakukan ekstraksi objek *saliency* tersebut dari *background*-nya, yang hasil akhirnya adalah sebuah citra yang telah terpisah dari *background*-nya. Metode *Saliency* berbasis model *Spectral residual* merupakan salah satu cara untuk melakukan proses tersebut. Metode ini menggunakan pendekatan nilai *spectral residual* untuk menentukan daerah yang menjadi *saliency*.

Untuk melakukan percobaan deteksi *saliency* seseorang harus melewati beberapa kesulitan, diantaranya adalah membeli seperangkat komputer *high-end* dan harus tetap tinggal ditempat, sehingga harus meluangkan waktu dan menghabiskan biaya yang tidak sedikit. Dengan demikian untuk mengatasi hal tersebut, diperlukan adanya suatu metode yang dapat memudahkan untuk melakukan percobaan deteksi *saliency* yakni dengan cara memvisualisasikan deteksi tersebut secara mobile yang nantinya diharapkan dapat memberikan kemudahan tanpa terhalang oleh adanya batasan tempat, waktu, dan biaya.

1.2. Permasalahan

Adapun permasalahan pada penelitian ini adalah:

1. Dibutuhkannya deteksi *saliency* yang dapat dengan tepat menunjukkan letak *saliency* suatu citra.
2. Dibutuhkannya deteksi *saliency* pada perangkat bergerak.

3. Penerapan deteksi *saliency* baru sebatas pada hardware dengan spesifikasi tinggi.

1.3. Tujuan dan Manfaat

Penelitian ini memiliki tujuan untuk melakukan ekstraksi *saliency* suatu citra dengan menggunakan pendekatan *spectral residual*, dan mevisualisasikannya pada perangkat bergerak sehingga dapat diaplikasikan untuk keperluan lain, misalnya untuk *image re-targeting* atau *segmentation*. Manfaat dari penelitian ini adalah untuk mendeteksi pusat perhatian suatu objek secara *mobile* menggunakan perangkat bergerak.

1.4. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah:

1. Citra yang diolah untuk dideteksi adalah citra yang diambil dengan menggunakan kamera berformat jpeg.
2. Metode yang digunakan adalah metode *spectral residual* dari xiaodihou dan liqing zhang.

1.5. Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini dibagi menjadi lima bab, masing-masing bab dapat diuraikan sebagai berikut:

1. Bab I Pendahuluan
Bab ini berisi uraian tentang latar belakang, tujuan penelitian, batasan masalah, dan sistematika penulisan.
2. Bab II Dasar Teori
Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini.
3. Bab III Perancangan Sistem dan Implementasi
Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Untuk mendukung penjelasan sistem digunakan blok diagram agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk diimplementasikan pada pembuatan perangkat lunak.
4. Bab IV Pengujian dan Analisis

Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem dalam penelitian ini dan menganalisis hasil uji sistem. Serta menjelaskan dengan detail tentang seluruh proses yang dijalani dalam penelitian ini.

5. Bab V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

BAB 2

TEORI PENUNJANG

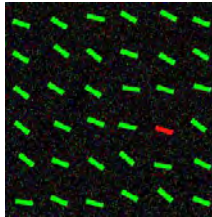
2.1. *Saliency*

Perhatian manusia tertarik terhadap rangsangan visual *saliency*. *Saliency* atau biasa disebut *salience* sangat penting bagi sistem biologis seperti manusia untuk cepat mendeteksi objek tertentu, baik itu yang menguntungkan seperti buah-buahan diantara pepohonan ataupun sesuatu yang mendatangkan bahaya seperti ular dalam pepohonan. Namun, mengidentifikasi tiap dan semua target yang menarik visual, seseorang membutuhkan kompleks komputasi yang tinggi sehingga menjadikannya sebagai sesuatu yang sulit dicapai meskipun untuk otak manusia yang paling canggih.

Ketidakmampuan untuk sepenuhnya memproses semua lokasi secara paralel menghasilkan masalah, jika hanya bisa memproses satu wilayah atau objek pada satu waktu, bagaimana seseorang harus memilih target perhatian? *Saliency* membantu otak memilih bagian yang penting dari target perhatian secara efisien. Otak manusia telah berkembang dengan cepat mendeteksi *saliency* secara otomatis dan realtime di seluruh bagian citra alam atau objek yang dilihat.

Saliency dibedakan menjadi 2, yaitu *bottom-up* dan *top-down*. *Bottom-up saliency* (faktor kepekaan) adalah ketika melihat bagian tertentu dari suatu citra yang bersifat lebih mencolok, sehingga bagian tersebut menjadi menarik perhatian. Sedangkan *top-down* (faktor keinginan user) adalah ketika sudah ditentukan terlebih dahulu bentuk ataupun warna benda yang dicari berdasarkan keinginan pelaku, sehingga yang dideteksi hanya benda yang memiliki bentuk atau warna yang sama. Pendekatan *saliency* paling banyak menggunakan metode *bottom-up*, dikarenakan sulitnya mengukur aspek internal suatu organisme, meskipun kadang kala penyebaran *bottom-up* terhadap visual *saliency* yang dianggap penting dapat dipengaruhi atau bahkan terkadang diganti oleh *top-down*. Contoh *bottom-up saliency* adalah objek merah sendirian di antara objek hijau akan menonjol dan akan

menarik perhatian dengan cara bottom-up (Gambar 2.1). sedangkan contoh top-down *saliency* adalah keinginan untuk melihat mainan anak dengan bungkusan bergambar naga merah, disampingnya terdapat bungkusan dengan banyak warna yang hidup seperti biru, hijau, kuning, tetapi tidak ada satu warna yang dianggap penting oleh perhatian sampai keinginan top-down untuk menemukan objek merah terpenuhi.[1]



Gambar 2.1 Contoh objek *saliency*. [1]

2.2. Log Spektrum

Dalam mencari fitur dari suatu citra, skala rasio tetap invarian adalah yang paling terkenal dan paling banyak digunakan [8, 9]. Sebuah skala rasio tetap invarian sistem pengukurannya akan menghasilkan format yang sama dari variabel-variabel yang digunakan dan hasil yang didapatkan akan sama juga. Sebagai contoh pertimbangan hal berikut:

$$X' = cx \quad (2.1)$$

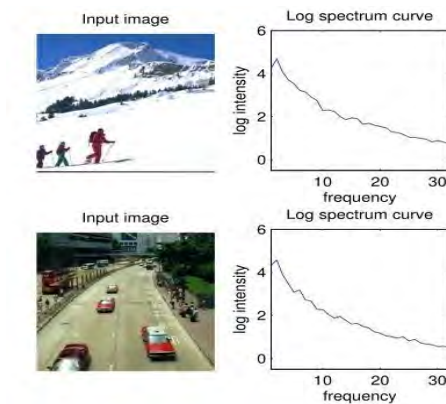
Jika X mewakili semua titik pada skala tertentu, dan titik masing-masing dikalikan dengan konstanta c , skala X yang dihasilkan juga akan menjadi skala rasio. Jika $A(f)$ adalah Amplitudo frekuensi dari suatu gambar, maka rata-rata Fourier spektrum dari citra alami mematuhi distribusi :

$$E\{A(f)\} \propto 1/f \quad (2.2)$$

Log Spektrum dapat diperoleh dengan

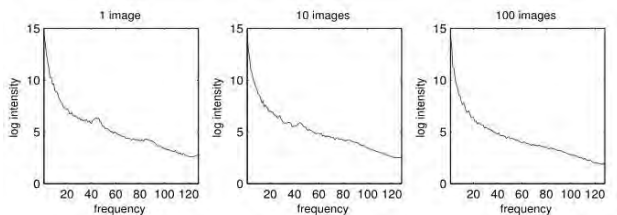
$$L(f) = \log(A(f)) \quad (2.3)$$

Pemodelan log spektrum telah banyak digunakan untuk analisa statistik citra alami [4,5,6,7].



Gambar 2.2 Kurva log spektrum

Gambar 2.2 Menunjukkan bahwa Log spektrum dari citra berbeda bisa menghasilkan kemiripan kurva yang sama.



Gambar 2.3 Kurva spektrum 1, 10, dan 100 citra

Gambar 2.3 menunjukkan kurva dengan input gambar 1, 10, 100 gambar yang diambil nilai rata-rata log spektrumnya mempunyai linieritas kurva yang hampir mirip.

2.3. Spectral residual

Dalam menganalisa fitur dari suatu citra alami dengan menggunakan metode statistika, kemiripan menyatakan redundansi sehingga untuk sistem yang ingin menyingkirkan informasi visual yang kurang berguna maka haruslah sistem tersebut mampu menyadari kemiripan dari inputan yang diberikan. Oleh karena itu, didalam perbedaan bentuk kurva log spektrum yang bisa diamati, apa yang perlu diperhatikan adalah spektrum yang memperlihatkan anomali yaitu *spectral residual*, yang diyakini sebagai tempat munculnya proto-objek. Sebuah prinsip dasar dalam sistem visual adalah untuk menekan respon terhadap fitur yang sering terjadi, sementara pada saat yang sama tetap peka terhadap fitur yang menyimpang dari norma, Oleh karena itu, hanya sinyal menyimpang yang dapat dikirim ke tahap pengolahan selanjutnya. *Spectral residual* didapat dengan menggunakan hipotesis Barlow[3] untuk menghilangkan fitur yang dianggap kurang berguna. Dimana $H(\text{Citra}) = H(\text{Inovasi}) + H(\text{Informasi Sebelumnya})$, $H(\text{Inovasi})$ menunjukkan bagian hal yang baru, dan $H(\text{Informasi Sebelumnya})$ adalah informasi berlebihan yang harus ditekan oleh sistem pengkodean. Jika informasi log spektrum $L(f)$ diperoleh sebelumnya, maka

$$H(R(f)) = H(L(f) | A(f)) \quad (2.4)$$

Dimana $A(f)$ bentuk umum log spektrum yang memberikan informasi yang lama yang ingin ditekan. $R(f)$ adalah *spectral residual*.

Pada gambar 2.3 terlihat bahwa rata-rata kurva menunjukkan fungsi linear yang hasilnya bisa dijadikan acuan filter untuk mendekati bentuk citra input. $A(f)$ bisa didapatkan dengan mengonvolusikan :

$$A(f) = h_n(f) * L(f) \quad (2.5)$$

Dimana h_n adalah matrik berukuran $n \times n$ yang didefinisikan :

$$h_n(f) = \frac{1}{n^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (2.6)$$

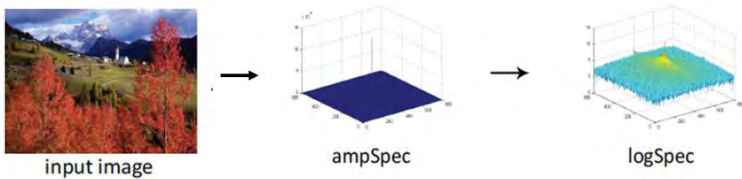
Maka *spectral residual* bisa didapatkan dengan :

$$R(f) = L(f) - A(f) \quad (2.7)$$

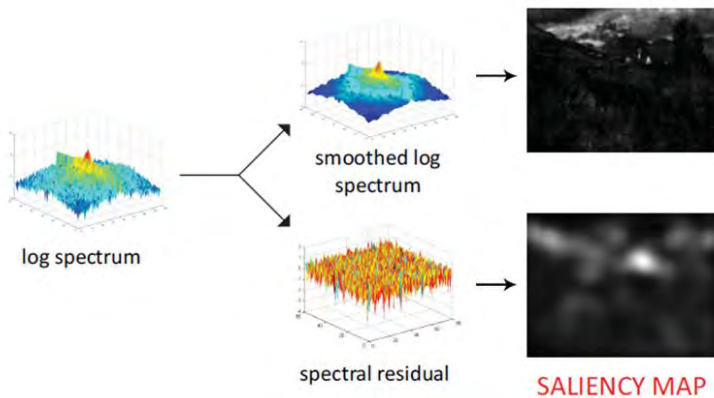
2.4. Saliency Map

Dalam pengolahan citra, *Saliency map* merupakan hasil dari penyusunan bagian yang merepresentasikan visual *saliency* pada suatu citra. *Saliency map* didapatkan setelah dilakukan pengelompokan terhadap bagian yang dianggap penting dan membuang bagian yang tidak penting.

Contoh pada gambar 2.4 , proses untuk mendapatkan *saliency map* pada citra dengan menggunakan *spectral residual*.



a)

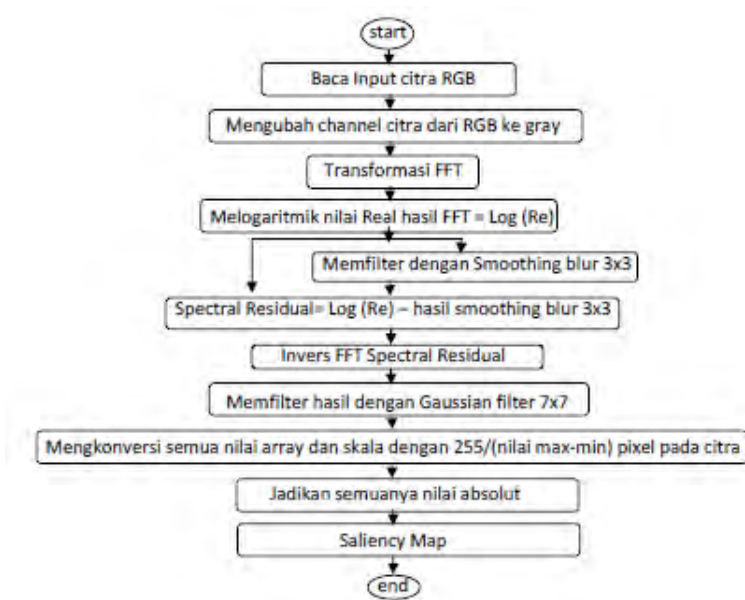


b)

Gambar 2.4 Proses mendapatkan *saliency map* dari suatu citra

a) Proses mendapatkan log spektrum dari citra b) Proses menghasilkan *saliency map*

Diagram alir untuk mendapatkan *saliency map* adalah sebagai berikut



Gambar 2.5 Diagram alir *saliency map*

2.5. Transformasi Fourier

Transformasi Fourier adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekuensi. Transformasi Fourier banyak digunakan untuk memindahkan domain dari suatu fungsi atau obyek ke dalam domain frekuensi. Di dalam pengolahan citra digital, transformasi fourier digunakan untuk mengubah domain spasial pada citra menjadi domain frekuensi. Analisa-analisa dalam domain frekuensi banyak digunakan seperti filtering. Dengan menggunakan transformasi fourier, sinyal atau citra dapat dilihat sebagai suatu obyek dalam domain frekuensi.

2.5.1. Transformasi Fourier kontinu

Transformasi Fourier kontinu 1D dari suatu fungsi waktu $f(t)$ didefinisikan dengan:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt \quad (2.8)$$

dimana : $F(\omega)$ adalah fungsi dalam domain frekuensi

ω adalah frekuensi radial $0 - 2\pi f$,

Transformasi Fourier kontinu 2D dari suatu fungsi spasial $f(x,y)$ didefinisikan dengan:

$$F(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot e^{-j(\omega_1 x + \omega_2 y)} dx dy \quad (2.9)$$

dimana $F(\omega_1, \omega_2)$ adalah fungsi dalam domain frekuensi

$f(x,y)$ adalah fungsi spasial atau citra

ω_1 dan ω_2 adalah frekuensi radial $0 - 2\pi f$.

Transformasi fourier yang digunakan dalam pengolahan citra digital adalah transformasi fourier 2D. Transformasi fourier semacam ini disebut continuous fourier transform, dan sulit dikomputasi karena ada operasi integral dan sifat kontinunya itu sendiri.

2.5.2. Transformasi Fourier Diskrit

Transformasi fourier diskrit atau disebut dengan Discrete Fourier Transform (DFT) adalah model transformasi fourier yang dikenakan pada fungsi diskrit, dan hasilnya juga diskrit. DFT 1D didefinisikan dengan :

$$F(k) = \sum_{n=1}^N f(n) \cdot e^{-j2\pi kn T / N} \quad (2.10)$$

DFT semacam ini banyak digunakan dalam pengolahan sinyal digital.

Transformasi Fourier Diskrit 2D adalah tranformasi fourier diskrit yang dikenakan pada fungsi 2D (fungsi dengan dua variabel bebas), yang didefinisikan sebagai berikut :

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} f(n_1, n_2) \cdot e^{-j2\pi f(k_1 n_1 / N_1 + k_2 n_2 / N_2)} \quad (2.11)$$

DFT 2D ini banyak digunakan dalam pengolahan citra digital, karena data citra dinyatakan sebagai fungsi 2D.

2.6. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar open source untuk perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. Dalam pembuatan aplikasi android penulis menggunakan *IDE* berupa Eclipse. Untuk dapat mengakses resource dan melakukan komputasi pengolahan citra pada ponsel maka digunakanlah tambahan *library* lain diantaranya Android NDK dan OpenCV.

2.6.1. OpenCV

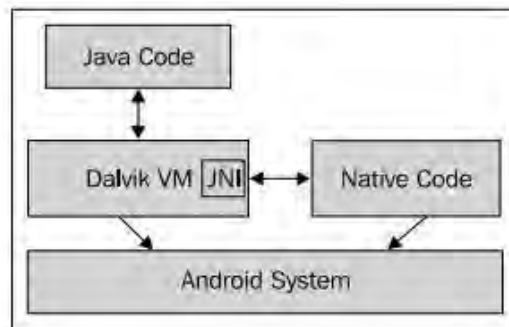
OpenCV (*Open Source Computer Vision Library*) adalah library yang dulunya dikembangkan oleh Intel dan sekarang didukung oleh Willow Garage yang bermain pada bidang real-time image processing. OpenCV gratis untuk digunakan dibawah lisensi Open Source BSD, baik untuk penggunaan akademis maupun komersil. OpenCV dapat dipakai dalam bahasa C++, C, Python dan Java serta mendukung antarmuka mendukung Windows, Linux, Mac OS, iOS dan Android. OpenCV didesain untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi *real-time* dan dioptimalkan dengan C/C++. *Library* dapat memanfaatkan pengolahan multi-core.



Gambar 2.6 Contoh aplikasi OpenCV pada android

2.6.2. JNI

JNI (*Java Native Interface*) adalah *framework* atau kerangka pemrograman yang memungkinkan kode Java berjalan di Dalvik Virtual Machine (VM) untuk berinteraksi dengan *library* yang ditulis dalam bahasa lain seperti C, C++ dan *assembly*.



Gambar 2.7 Prinsip kerja java dan native c di android

Aplikasi Android dengan kode native biasanya memiliki struktur standar proyek di mana terdapat folder “jni” yang berisi kode sumber native dan build file Android.mk yang sesuai.

2.7. Smoothing

Smoothing atau biasa disebut *blurring* adalah proses mengaburkan gambar untuk mendapatkan hasil yang lebih baik dengan mengurangi *noise*. *Smoothing* menggunakan bermacam-macam ukuran kernel filter untuk diaplikasikan pada proses pengolahan gambar.

Ada beberapa metode *smoothing* diantaranya adalah :

1. Blur : *smoothing* gambar dengan menggunakan kernel sebesar dimensi pixel gambar tersebut

$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ & & \dots & & & \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix} \quad (2.12)$$

2. Gaussian : *smoothing* dengan menggunakan kernel gaussian

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.13)$$

2.8. Precision dan Recall

Dalam dunia pengenalan pola (*pattern recognition*) dan temu kembali informasi (*information retrieval*), *precision* dan *recall* adalah dua perhitungan yang banyak digunakan untuk mengukur kinerja dari sistem / metode yang digunakan. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

Dalam menyusun nilai *Precision* dan *Recall*, maka diperlukan lah komponen penyusun lainnya seperti *TP*(*True Positive*), *FP*(*False Positif*), *FN*(*False Negative*), dan *TN*(*True Negative*). *TP* adalah persentase kejadian dimana data hasil uji memberikan pernyataan benar sesuai dengan data pembanding yang sudah valid yang juga memberikan pernyataan benar. *FP* adalah persentase kejadian dimana data hasil uji memberikan pernyataan benar pada suatu kejadian padahal data valid sebagai pembanding memberikan pernyataan salah. *FN* adalah persentase kejadian dimana data hasil uji memberikan pernyataan salah padahal data valid memberikan pernyataan yang benar. *TN* adalah

persentase kejadian dimana data hasil uji memberikan pernyataan salah, data valid juga memberikan pernyataan salah.

Secara umum *precision* dan *recall* dapat dirumuskan sebagai berikut:

$$precision = \frac{TP}{TP + FP} \quad (2.14)$$

$$recall = \frac{TP}{TP + FN} \quad (2.15)$$

BAB 3

DESAIN DAN IMPLEMENTASI

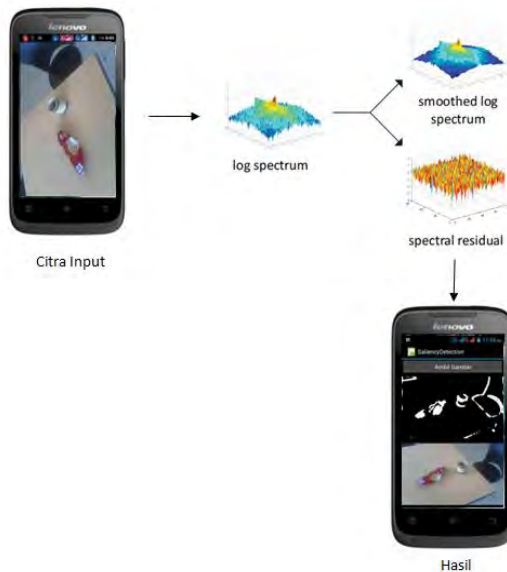
3.1. Desain Sistem

Untuk menjelaskan kerangka kerja sistem secara keseluruhan, maka desain sistem dibagi menjadi dua bagian yaitu:

1. Alur Kerja Sistem
Dalam hal ini akan dijelaskan secara umum aplikasi yang akan dibuat.
2. Prinsip Kerja Sistem
Dalam hal ini akan dijelaskan cara kerja pembuatan aplikasi hingga penggunaan aplikasi.

3.1.1. Alur Kerja Sistem

Sistem dirancang untuk menerapkan deteksi *saliency* dalam bentuk aplikasi perangkat bergerak. Alur kerja sistem secara keseluruhan digambarkan pada gambar 3.1.



Gambar 3.1 Alur kerja sistem secara umum

Deteksi *Saliency* dibuat dalam sebuah paket aplikasi menggunakan bahasa pemrograman java dan C native pada android. Hal yang pertama dilakukan adalah saat program dijalankan, program akan memicu kamera yang ada pada perangkat agar menyala. Kedua, untuk mengambil citra yang akan diproses, disediakan *buffer* pada program untuk menyimpan data citra sementara pada sdcard perangkat. Jika pengguna aplikasi tidak menginginkan citra yang sudah terlanjur diambil maka pengguna cukup menekan tombol x yang kemudian program akan secara otomatis membuang data citra lama dan men-trigger kamera kembali untuk mengambil citra yang baru. Untuk mengambil citra, pengguna harus menekan tombol centang setelah pemotretan.

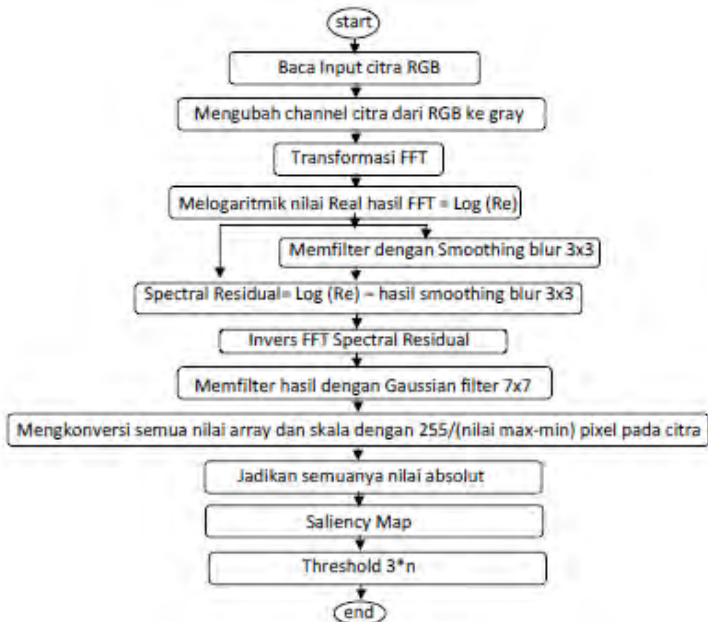


Gambar 3.2 Tampilan aplikasi ketika ingin mengambil suatu citra

Ketiga, gambar akan diproses dengan menggunakan algoritma *saliency* yang sebelumnya sudah dimasukkan kedalam bundle aplikasi. Dalam buku ini penulis menggunakan metode *spectral residual* untuk mendapatkan *saliency map* dari gambar masukan. Detail proses *saliency* akan dijelaskan lebih lanjut pada sub bab prinsip kerja sistem.

3.1.2. Prinsip Kerja Sistem

Berdasarkan konsep sistem yang telah dijelaskan pada sub bab 3.1.1, pada sub bab 3.1.2 ini akan dijelaskan tentang prinsip kerja sistem dari awal hingga akhir. Secara keseluruhan sistem mengikuti prinsip seperti diagram alur pada gambar 3.3



Gambar 3.3 Diagram Alur Prinsip Kerja Sistem

Proses pengambilan citra dari kamera perangkat cukup mudah, karena sudah disediakan library dari java untuk memberikan cara bagaimana mengakses *hardware* dari perangkat mobile berplatform android.

Proses kedua adalah pengkonversian channel citra masukan dari kamera dari rgb ke single channel menjadi gray untuk mempermudah pengkodean dalam memproses citra tanpa menghilangkan esensi dari citra itu sendiri.

Proses ketiga adalah mentransformasi citra dengan Fast Fourier Transform. Nilai real dari transformasi ini dilogaritmik dan di filter dengan *smoothing blur* 3x3. Hasil dari pengurangan log nilai real spektrum dikurangi dengan log nilai real sesudah spektrum difilter dengan *smoothing blur* 3x3 akan menghasilkan model *Spectral residual*. *Spectral residual* kemudian ditransformasikan dengan Invers FFT untuk menggabungkan nilai real dan imajiner menjadi 1 channel warna. Untuk

hasil tampilan *saliency map* yang bagus maka digunakanlah filter *Gaussian* dengan ukuran kernel 7×7 .

3.2. Implementasi Sistem

Diperlukan beberapa tahapan untuk bisa mendapatkan *saliency* sebuah citra pada android. Implementasi sistem merupakan gabungan dari proses ekstraksi fitur pada citra untuk menghasilkan log spektrum yang kemudian diproses menggunakan transformasi fourier dan difilter dengan filter *gaussian* untuk menghasilkan sebuah model. Tahapan-tahapan yang harus dilakukan adalah :

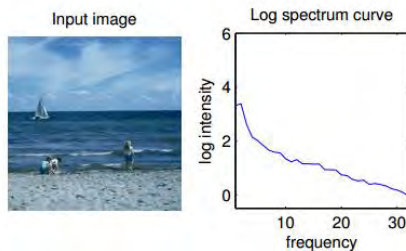
1. Penyusunan program
2. Deteksi *saliency* pada android

3.2.1. Penyusunan Program

Penyusunan program dibagi menjadi beberapa bagian yaitu :

- a. Ekstraksi citra log spektrum dari kamera

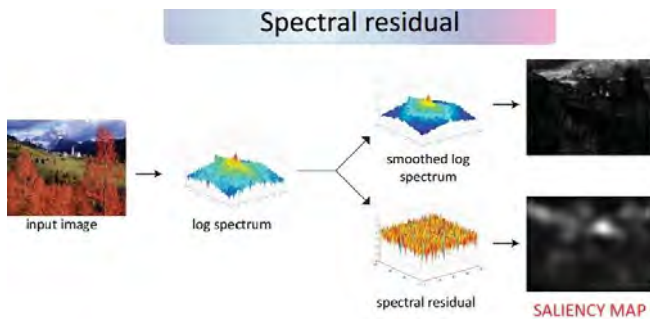
Ekstraksi citra log spektrum dibuat dengan menggunakan library opencv pada android, citra input digrayscalekan terlebih dahulu, kemudian diubah ukurannya menjadi $\frac{1}{4}$ dari ukuran pixel kamera perangkat. Ini dikarenakan penulis telah melakukan ujicoba di beberapa perangkat tanpa mengubah ukuran, hasil *saliency map*nya berupa gambar buram total atau *blank*. ukuran pixel bisa diubah menjadi berapapun untuk mendapatkan hasil deteksi *saliency* yang terbaik. Dalam percobaan, penulis mendapatkan hasil yang paling baik pada saat ukuran citra diubah menjadi $\frac{1}{4}$ dari ukuran pixel kamera perangkat. Nilai log spektrum diperoleh dengan melogaritma tiap-tiap nilai pixel pada citra dengan menggunakan fungsi `core.log()` pada native c android.



Gambar 3.4 Visualisasi log spektrum sebuah citra

b. Model *Spectral residual* dan *saliency map*

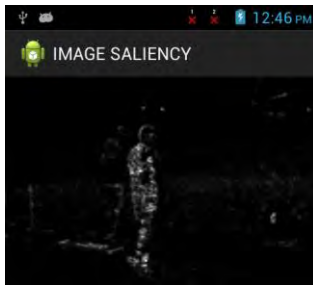
Setelah citra berhasil diekstrak menjadi log spektrum, barulah model *spectral residual* bisa diperoleh dengan mengurangi nilai log amplitudo spektrum dengan log amplitudo spektrum yang telah difilter dengan smoothing blur 3x3 seperti pada persamaan 2.7. filter smoothing blur ini diperlukan untuk mencari informasi sinyal kurva yang menyimpang pada citra, yang akan menjadi bagian dari proto-objek yang dideteksi. Informasi ini yang dianggap penting.



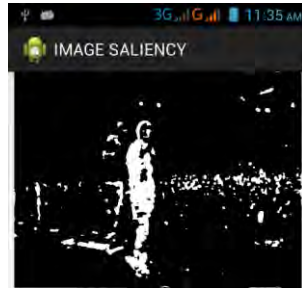
Gambar 3.5 *Spectral residual*

c. Deteksi *Saliency* pada objek

Deteksi *saliency* pada penelitian ini merupakan suatu program untuk mendeteksi *saliency* dalam citra yang diambil oleh kamera perangkat. *Saliency map* didapat dari invers transformasi fourier pada model *spectral residual* yang hasilnya masih berupa gambar *grayscale*. Untuk lebih memperjelas hasil *saliency* maka hasil Invers transformasi fourier tersebut harus difilter terlebih dahulu dengan filter gaussian 7x7 dan dithreshold dengan nilai 3x rata-rata dari jumlah nilai pixel *saliency map* seperti yang ditunjukkan pada gambar 3.6. hasil *saliency* yang telah dithreshold akan digunakan untuk mendapatkan nilai akurasi dengan membandingkan kemiripan yang terjadi antara citra hasil uji yang telah dithreshold dengan data dari *ground truth* pada citra tersebut. Proses pengukuran akurasi ini akan dijelaskan lebih lanjut pada sub bab 3.2.3.



a)

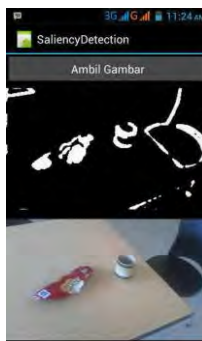


b)

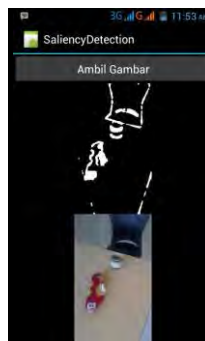
Gambar 3.6 Hasil deteksi objek *saliency* pada perangkat
a) Citra *grayscale* b) Citra *grayscale* setelah dithreshold

3.2.2. Deteksi *Saliency* Pada Android

Dalam mengimplementasikan hasil dari penyusunan program, maka diperlukanlah perangkat yang mendukung, salah satunya adalah handphone berplatform android. Program disusun dalam bentuk sebuah aplikasi yang bisa dijalankan pada android. Dalam pengambilan citra input posisi kamera harus berada dalam kondisi *landscape*, hal ini diperlukan untuk mendapatkan hasil visualisasi yang baik pada perangkat android karena setting tampilan layout output *saliency* masih menyesuaikan gambar inputan dan ukuran layar tipe perangkat yang digunakan. Terlihat pada gambar 3.7 bahwa mode portrait citra input dan *saliency* kelihatan mengecil menyesuaikan layar perangkat.



a)

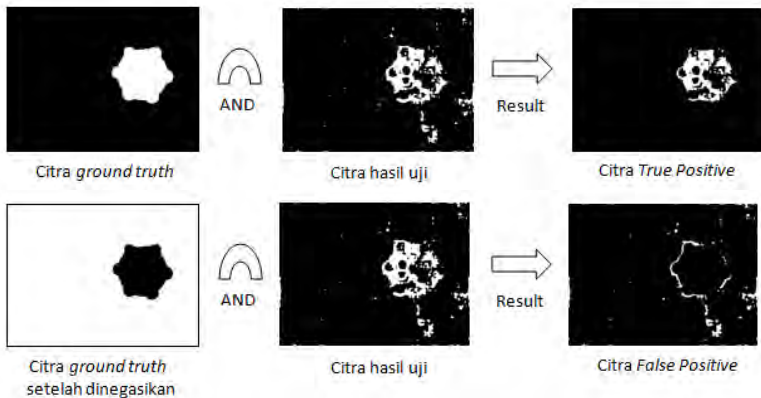


b)

Gambar 3.7 Hasil deteksi *saliency*
a) Mode landscape; b) Mode portrait

3.2.3. Perhitungan evaluasi model deteksi

Perhitungan evaluasi model pada penelitian ini menggunakan nilai *Precision* dan *recall* dari data set *paper* Hou[2] dan Achanta[10]. *Precision* menyatakan persentase dari jumlah pixel *salient* hasil uji yang sesuai dengan data pixel *salient* dari *ground truth* dibanding dengan jumlah pixel *salient* yang terdeteksi oleh model, *Recall* atau biasa disebut *Hit Rate* adalah persentase jumlah pixel *salient* dari hasil uji yang sesuai dengan jumlah pixel *salient* dari *ground truth*. Cara melakukan analisa perhitungan adalah dengan mencari nilai *TP*(*true positive*), *FP*(*false positive*), dan *FN*(*False Negative*). *TP* adalah nilai persentase ketepatan dari model deteksi *saliency* dalam mendeteksi objek *saliency* sesuai dengan data dari *ground truth*. *FP* adalah nilai banyaknya kejadian dari model deteksi *saliency* menganggap suatu objek adalah objek *saliency* padahal data dari *ground truth* tidak menunjukkan bahwa objek itu adalah objek *saliency*. Cara mendapatkan nilai *TP* adalah dengan menggunakan logika operator *AND* pada array citra dari *ground truth* dengan citra hasil uji kemudian jumlah pixel berwarna putih (bernilai 1) nya dihitung. Sedangkan untuk nilai *FP* bisa dicari dengan menegasikan citra *ground truth* terlebih dahulu kemudian hasil negasi tersebut dengan citra hasil uji digunakan operator *AND*.



Gambar 3.8 Teknik untuk mendapatkan citra *TP* (atas) dan *FP* (bawah).

Untuk nilai FN bisa didapat dengan mensubtraksi citra ground truth dengan citra hasil TP , kemudian hasilnya dinegasikan. ketika nilai TP , FP , FN diketahui, maka nilai $TN = 1-(TP+FP+FN)$.

3.3. Desain Sistem Survei

Desain sistem Survei menggunakan sistem kuisioner online pada *google docs*. Kuisioner ini bertujuan untuk mengecek kembali apakah deteksi *saliency* pada sistem sudah sesuai dengan yang diharapkan, dengan mencocokkan hasil pandangan *saliency* menurut masyarakat dan hasil *saliency* pada perangkat android. Peserta survei terdiri dari mahasiswa akademisi, praktisi dan umum.

Kuisioner Survei:

https://docs.google.com/forms/d/1YzIEeMxNw3bsUpvx_NMBDMGkoHa6FNhB-DWIPEhrv2E/viewform.

Kuisioner berisi pertanyaan yang sama pada tiap citra yang ditampilkan. "Manakah Objek yang paling menarik perhatian Anda?". Jawaban berupa checklist pilihan ganda, isi dari jawaban sebelumnya telah disesuaikan dengan citra objek yang memungkinkan untuk menjadi objek *saliency* sehingga para partisipan survei tidak kesulitan untuk segera memilih jawaban sesuai dengan objek yang menarik perhatian visual mereka. Jawaban boleh lebih dari satu karena memungkinkan dalam 1 citra beberapa objek bisa menarik perhatian partisipan.

Ada 10 citra yang dijadikan objek survei deteksi *saliency*. Kesemuanya diambil dari website Liqing Zhang dan Xiaodi Hou dimuat, Ke 10 citra tersebut dianggap telah mewakili untuk 62 dataset yang ada karena keberagaman objek telah dipilih sesuai dengan hasil dari deteksi *saliency* yang telah dicapai.

Berikut contoh form survei yang digunakan,

Survei Visi Komputer

Survei ini bertujuan untuk mengumpulkan data mengenai pusat perhatian manusia terhadap sesuatu yang dilihatnya, terimakasih atas partisipasinya.

* Wajib

Citra 01











Pada citra 01, manakah objek yang paling menarik perhatian menurut anda *

- ☐ rusa disebelah kanan
- ☐ binatang putih sebelah kiri
- ☐ 3 rusa yang dibawah pohon
- ☐ bangunan berwarna kuning
- ☐ Yang lain:

Gambar 3.9 Contoh Form Survei Citra Pertama

Selanjutnya, form survei akan menampilkan citra berikutnya yang mengikuti format sama dengan yang sebelumnya sampai dengan citra ke 10. Dibawah adalah citra-citra yang dijadikan survei.

No	Citra	Citra
1-2		
3-4		
5-6		
7-8		

No	Citra	Citra
9-10		

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini akan dijelaskan mengenai pengujian dan analisa hasil pengujian aplikasi yang telah diimplementasikan apakah telah sesuai dengan desain sistem yang telah dirancang dan memberikan hasil yang diharapkan. Adapun pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian Aplikasi Pada Android dan Komputer
2. Analisa Hasil Survei

4.1. Pengujian Aplikasi

Pada sistem yang telah dibuat, pengujian dilakukan dengan menggunakan handphone lenovo a369i dan asus zenfone 4. Tabel 4.1 dan tabel 4.2 menunjukkan spesifikasi dari perangkat yang digunakan.

Tabel 4-1 Spesifikasi perangkat uji android lenovo a369i

Komponen	Spesifikasi
OS	Android OS, v4.2 Jelly Bean
Chipset	Mediatek MT6572
CPU	1,3 GHz Cortex Dualcore
GPU	Mali-400
Ukuran Layar	480 x 800 pixels,
Memori	512 MB
Kamera	2MP, 1280x768 pixels, autofocus

Tabel 4-2 Spesifikasi perangkat uji android zenfone 4

Komponen	Spesifikasi
OS	Android OS, v4.3 Jelly Bean
Chipset	Intel Atom Z2520
CPU	1,2 GHz Dualcore
GPU	PowerVR SGX544MP2
Ukuran Layar	480 x 800 pixels,
Memori	1024 MB
Kamera	5MP, 2592x1944 pixels, autofocus

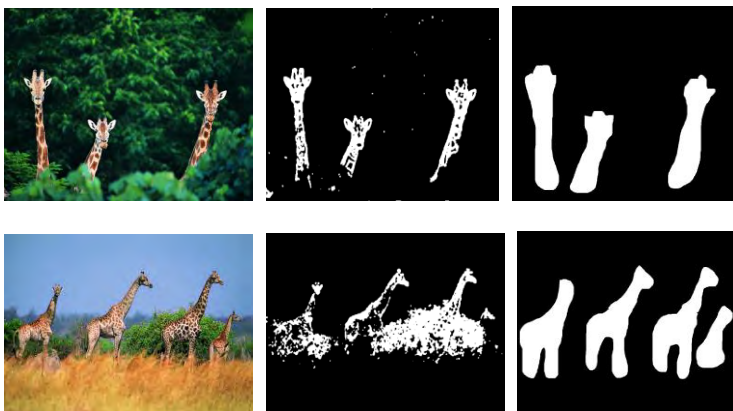
Dan sebagai pembanding, dilakukan pengujian aplikasi pada komputer dengan spesifikasi seperti yang tertera pada tabel 4.3.

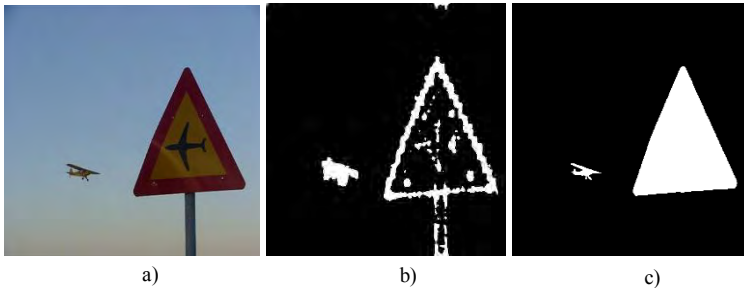
Tabel 4-3 Spesifikasi perangkat uji komputer

Komponen	Spesifikasi
OS	Linux Debian Squeeze 6.0
CPU	Intel T4500 2,2GHz Dualcore
GPU	Intel GMA4500 Accelerator
Ukuran Layar	1366x768
Memori	2GB

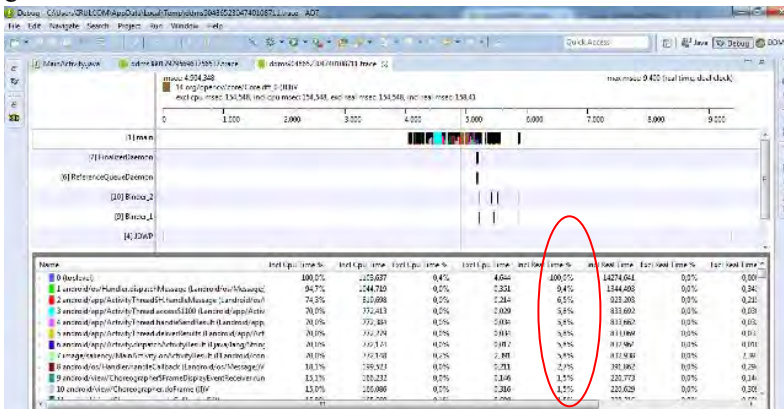
4.1.1. Pengujian *Saliency* Pada Android

Pada pengujian ini, dilakukan pengujian dengan menggunakan 73 citra yang terdiri dari 58 dataset oleh Xiaodi Hou[2] dan 15 dataset oleh Achanta[10]. Dari 73 citra tersebut 3 diantaranya tidak dapat diolah perangkat android lenovo a369i, sedangkan perangkat zenfone 4 masih bisa mengolah 3 citra tersebut dengan rentang waktu 4,6 sampai 6,6 detik. Hasil yang bagus terjadi ketika objek *saliency* warnanya lebih kontras dibanding bagian-bagian tetangganya seperti pada citra “42.jpg”, kepala jerapah kecoklatan diantara dedaunan hijau (gambar 4.1). Berturut-turut ditampilkan citra nomor 42, 41 dan 63.





Pengujian performansi dilakukan dengan menggunakan tool DDMS pada eclipse. DDMS merupakan tool profiling bawaan eclipse yang cukup powerful untuk melakukan uji performansi seperti nilai performansi beban waktu yang diperlukan tiap *function* untuk melakukan tugasnya, seperti yang ditunjukkan bulatan merah pada gambar 4.2.



Dari Pengujian yang dilakukan terhadap performa perangkat diperoleh data waktu yang cukup signifikan hingga 6000an milisekon untuk memproses citra input menjadi citra *saliency*, seperti yang

ditunjukkan pada tabel 4-4. Waktu pemrosesan tercepat terjadi pada citra “60.jpg” yang mempunyai ukuran 256x256 pixel dengan waktu sebesar 256ms pada lenovo a369i dan 315ms pada asus zenfone4(warna hijau), sedangkan waktu pemrosesan terlama terjadi pada citra “58.jpg” yang mempunyai ukuran 1280x1239 pixel dengan waktu sebesar 6618ms pada asus zenfone4. Perangkat lenovo a369i membutuhkan waktu terlama pada citra “13.jpg” yang berukuran 800x600 pixel dengan waktu sebesar 3918ms(warna kuning) dan tidak bisa memproses pada tiga citra berukuran lebih besar atau sama dengan 1280x960 pixel(warna merah).

Tabel 4-4 Hasil perbandingan waktu deteksi *saliency* perangkat

Nomor Citra	Ukuran	waktu(ms)		
		komputer	Lenovo a369i	Asus zenfone 4
42	800x531	210	1439	1537
41	800x534	250	1196	1537
67	400x300	130	1271	1258
60	256x256	20	256	315
13	800x600	160	3918	1979
57	1280x1280	640	X	6604
58	1280x1239	910	X	6618
61	1280x960	460	X	4605

Hasil uji akurasi model detektor dalam mendeteksi objek *saliency* pada 73 dataset uji adalah sebesar 84,1% seperti yang ditampilkan oleh tabel 4-5. Namun akurasi model sebagian besar dipengaruhi oleh nilai TN, yaitu sukses mendeteksi bagian yang bukan objek *saliency* pada citra. Padahal model detektor *saliency* diharapkan dapat mendeteksi pada objek yang *saliency*, bukan keberhasilan dalam mendeteksi objek yang bukan *saliency*. Secara umum model masih sangat jauh dari ideal. Tingkat *precision* dan *recall* juga kurang bagus karena masih dibawah 80%, padahal idealnya *precision* dan *recall*

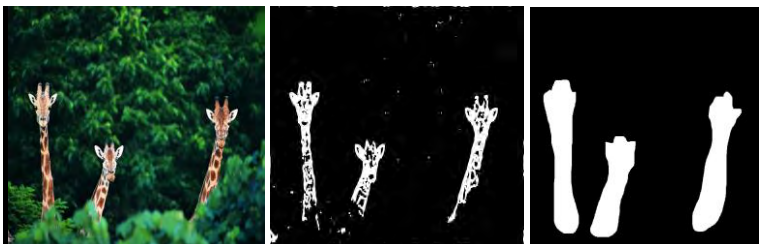
minimal harus diatas nilai 80%. ini berarti sebagian besar pixel *salient* hasil uji banyak yang tidak sesuai atau *miss* dengan data pixel *salient* dari *ground truth*.

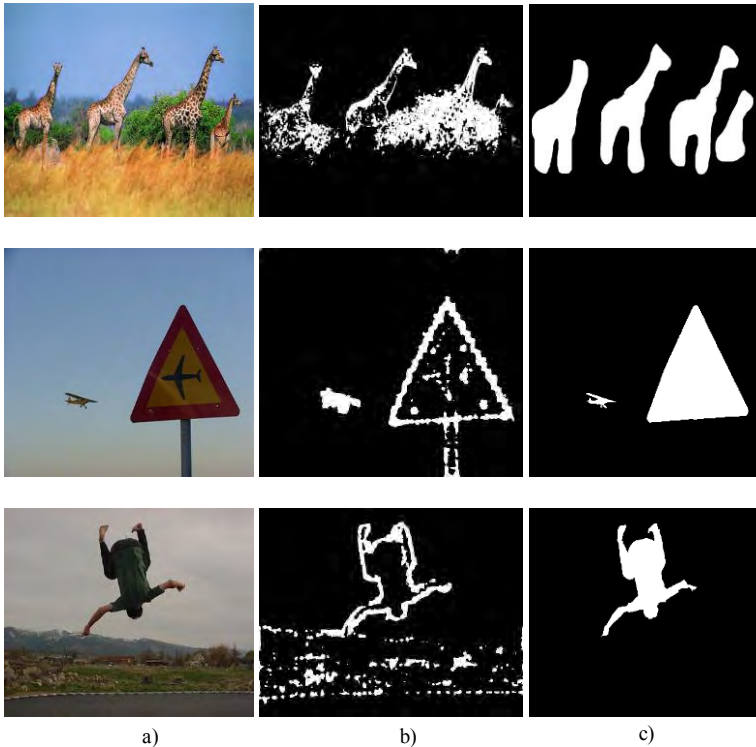
Tabel 4-5 Performansi hasil uji model pada dataset

Parameter	Max	min	Avg
TP	11.13%	0.07%	3.50%
FP	11.42%	0.34%	4.58%
FN	44.35%	0.29%	11.31%
TN	96.16%	47.45%	80.61%
<i>Precision</i>	95.85%	2.12%	42.74%
<i>Recall</i>	82.69%	3.29%	26.94%
Accuracy	96.69%	55.31%	84.11%

4.1.2. Pengujian *Saliency* Pada Komputer

Pada pengujian ini, dilakukan pengujian yang sama pada android, yaitu 73 citra yang terdiri dari 58 dataset oleh Xiaodi Hou[2] dan 15 dataset oleh Achanta[10]. Hasil deteksi objek *saliency* yang bagus pada komputer sama dengan hasil pengujian pada android yaitu terjadi ketika objek *saliency* warnanya terlihat lebih kontras dibanding bagian-bagian tetangganya seperti pada citra “42.jpg”, kepala jerapah kecoklatan diantara dedaunan hijau.




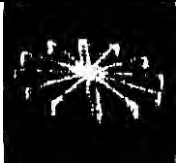









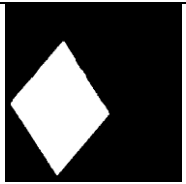






Gambar 4.3 Contoh Hasil Deteksi *Saliency* pada komputer
a) Citra asli b) Citra *saliency* c) Citra *ground truth*

Dari Pengujian yang dilakukan terhadap performa deteksi *saliency* pada komputer diperoleh data waktu yang lebih kecil hampir 10x lipat daripada pengujian aplikasi yang dilakukan pada android. Waktu yang diperoleh berkisar pada angka 200-300ms. Waktu pemrosesan tercepat terjadi pada citra “60.jpg” yang mempunyai ukuran 256x256 pixel dengan waktu sebesar 20ms sama dengan citra tercepat yang terjadi pada pengujian di android dan waktu terlama terjadi pada citra “16.jpg” yang mempunyai ukuran 757x600pixel dengan waktu sebesar 2428ms.

Dari hasil kualitas gambar, hampir tidak ada perbedaan mencolok antara hasil output dari komputer maupun dari android seperti pada tabel 4-6.

Tabel 4-6 Perbandingan hasil *saliency* pada android dan komputer


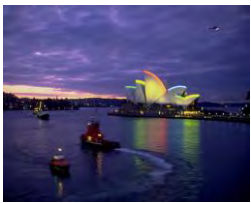

Citra asli	<i>Saliency</i> (komputer)	<i>Saliency</i> (android)	<i>Ground truth</i>
			
			
			
			

Untuk menganalisa kemiripan hasil antara deteksi *saliency* menggunakan komputer dengan perangkat bergerak, maka digunakanlah metode seperti pada subbab 3.2.3, data yang dibandingkan adalah data hasil uji *saliency* oleh komputer yang telah dithreshold dengan data hasil uji *saliency* oleh perangkat bergerak yang juga telah dithreshold sebelumnya.



4.2. Analisa Hasil Survei

Dari hasil survei didapat 65 responden aktif masyarakat dengan berbagai macam profesi. Dalam kuisioner ini responden akan memilih objek atau bagian dari 10 citra yang ditampilkan pada browser, 10 citra survei dipilih berdasarkan keberagaman objek yang ada pada citra sehingga cukup mewakili dari 62 citra yang ada pada dataset Hou[2]. Masing-masing responden akan memilih objek atau bagian mana yang menjadi pusat perhatian pada checklist yang telah disediakan, atau ketika mempunyai pendapat sendiri, tinggal mengisi checklist “lainnya” pada ke bagian paling bawah list untuk mengisi jawaban yang belum ada pada pilihan. Responden bisa memilih lebih dari satu pilihan jawaban jika dirasa objek yang dianggap penting lebih dari satu.

Tabel 4-7 Hasil survei masyarakat terhadap citra

No	citra	Pilihan jawaban	Jumlah
1.		<ul style="list-style-type: none"> - rusa disebelah kanan - binatang putih sebelah kiri -rusa yang dibawah pohon -bangunan berwarna kuning - Lainnya 	31 6 4 18 13
2.		<ul style="list-style-type: none"> - kapal merah besar di tengah - kapal merah kecil di sebelah kiri - kapal putih - bangunan sydney - helikopter - Lainnya 	8 5 0 55 5 8
3.		<ul style="list-style-type: none"> - kaos putih manusianya - tangan manusianya - 3 bebek didepan - bebek putih disebelah kanan - Lainnya 	26 6 14 10 17

No	Citra	Pilihan jawaban	Jumlah
4.		<ul style="list-style-type: none"> -3 sapi hitam didepan - sapi putih sendirian di sebelah kanan - rumah - tanaman hijau disebelah kiri - Lainnya 	16 8 13 5 29
5.		<ul style="list-style-type: none"> - wajah singa tengah yang kelihatan taringnya - 2 singa disebelah kanan - wajah singa yang menganga ditengah - 4 singa - Lainnya 	40 3 12 6 7
6.		<ul style="list-style-type: none"> - jerapah yang melihat kearah pembaca - 3 jerapah yang tinggi - jerapah yang kecil di pojok kanan - 4 jerapah - Lainnya 	11 9 5 26 19
7.		<ul style="list-style-type: none"> - 3 wajah jerapah - jerapah yang disebelah kiri - jerapah yang disebelah kanan - Lainnya 	41 6 9 9
8.		<ul style="list-style-type: none"> - batu tinggi besar disebelah kanan - 3 payung diatas perahu - penumpang dibelakang perahu - batu disebelah kiri - Lainnya 	40 7 6 3 18

9.		<ul style="list-style-type: none"> - kepala botak - 2 orang penyanyi - kumpulan audience dibawah - kain putih disebelah kanan atas - Lainnya 	39 10 3 3 11
10.		<ul style="list-style-type: none"> - rumah besar ditengah - 2 rumah ditengah - kabut diatas rumah - gunung dibelakang rumah - Lainnya 	32 8 13 11 7

Dari tabel 4-7 terlihat bahwa terjadi keberagaman pandangan visual *saliency* antar seseorang, untuk citra nomor 1, rusa berwarna coklat di sebelah kanan bagian citra lebih dominan dengan mendapat 31 suara (47%) dari 65 total responden sedangkan bangunan berwarna kuning berada pada posisi kedua dengan 17 suara(26%). Hasil dari deteksi *saliency* pada android menunjukkan bangunan kuning dan pepohonan yang menjadi objek *saliency*. Sehingga algoritma detektor dianggap gagal untuk mencari objek *saliency*(hanya 26%). Pada citra nomor 2, bangunan sydney mendominasi dengan 55 suara(84%), disusul kapal merah besar dan lainnya, aplikasi berhasil mendeteksi bangunan sydney sebagai objek *saliency* dan kapal besar dan kecil, maka bisa diambil kesimpulan untuk citra nomor 2 algoritma detektor berhasil mendeteksi objek *saliency*. Untuk citra nomor 3,6,7 dan 9 masing-masing dengan urutan dominan suara 32(49%), 26(40%), 40(63%), 49(75%) sesuai dengan algoritma detektor. Untuk citra nomor 4,5, dan 10 aplikasi gagal dalam menentukan objek *saliency*, responden dengan pilihan berturut-turut, lainnya, objek singa, dan rumah besar ditengah menjadi paling dominan sedangkan menurut aplikasi, objek *saliency*nya berturut-turut adalah tanaman(7%), lainnya(10%), dan gunung(17%) . Untuk citra nomor 8, objek batu lebih menarik dibanding dengan objek manusia yang menaiki perahu menurut survei partisipan, aplikasi lebih memilih penumpang dalam perahu(18%) sebagai objek *saliency*. Rangkuman

hasil terdapat pada tabel 4-8. Untuk data *F-measure* didapat dari nilai kecocokan antara data dari *ground truth* dengan data dari hasil uji seperti yang ditunjukkan pada tabel 4-5.

Tabel 4-8 Rangkuman hasil survei terhadap aplikasi

nomor citra	vote dominan	vote(%)	model/ <i>ground truth</i> (%)
1	Tidak	26	70%
2	Ya	84	86%
3	Ya	49	80%
4	Tidak	7	73%
5	Tidak	10	55%
6	Ya	40	82%
7	Ya	63	90%
8	tidak	17	88%
9	ya	75	84%
10	tidak	18	84%

BAB 5

PENUTUP

5.1. Kesimpulan

Pada penelitian ini akan didapatkan kesimpulan berdasarkan hasil analisa dan pengujian sistem pada pembuatan deteksi *saliency* pada perangkat bergerak adalah sebagai berikut:

1. Objek yang warnanya lebih unik dan terang dibanding bagian sekitar akan lebih mudah dideteksi sebagai objek *saliency*.
2. Tingkat akurasi model *spectral residual* dalam mendeteksi *saliency* sebesar 84,1% sebagian besar dipengaruhi oleh kesuksesan mendeteksi objek yang bukan *saliency*.
3. Hasil output citra menunjukkan bahwa deteksi *saliency* pada android mempunyai hasil yang mirip dengan yang dilakukan pada komputer.
4. Hasil survei menunjukkan bahwa hasil deteksi *saliency* pada android menggunakan model *spectral residual* dengan objek *saliency* menurut partisipan survei rata-rata kesesuaiannya mencapai 62.2%.

5.2. Saran

Untuk pengembangan lebih lanjut ada beberapa saran mengenai penelitian ini sebagai berikut:

1. Beberapa algoritma deteksi *saliency* bisa digunakan dalam satu aplikasi di perangkat bergerak untuk benchmarking agar bisa mengatasi data bias pada dataset.
2. Perlunya performansi memori yang digunakan untuk mengecek kebutuhan ram
3. Citra survei yang digunakan lebih beragam dan partisipan survei ditambah untuk lebih mengakuratkan hasil.

LAMPIRAN

Kuesioner ini untuk mengetahui tingkat akurasi berdasarkan hasil deteksi *saliency* menurut survei masyarakat

Survei Visi Komputer

Survei ini bertujuan untuk mengumpulkan data mengenai pusat perhatian manusia terhadap sesuatu yang dilihatnya, terimakasih atas partisipasinya.

* Wajib

Citra 01



Pada citra 01, manakah objek yang paling menarik perhatian menurut anda *

- ☐ rusa disebelah kanan
- ☐ binatang putih sebelah kiri
- ☐ 3 rusa yang dibawah pohon
- ☐ bangunan berwarna kuning
- ☐ Yang lain:

Lanjutkan »

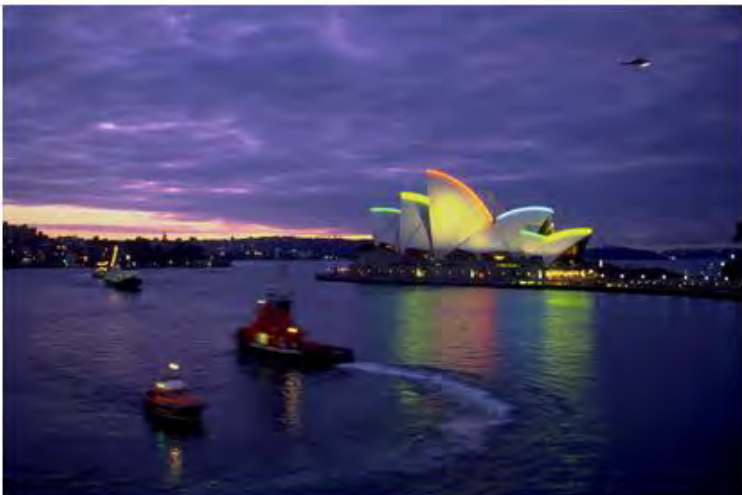
Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

*Wajib

Selanjutnya

Citra 02



Pada citra 02, manakah objek yang paling menarik perhatian menurut anda *

- ☐ kapal merah besar di tengah
- ☐ kapal merah kecil di sebelah kiri
- ☐ kapal putih
- ☐ bangunan sydney
- ☐ helikopter
- ☐ Yang lain:

[◀ Kembali](#)

[Lanjutkan ▶](#)

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

* Wajib

Selanjutnya

Citra 03



Pada citra 03, manakah objek yang paling menarik perhatian menurut anda *

- ☐ kaos putih manusianya
- ☐ tangan manusianya
- ☐ 3 bebek di depan
- ☐ bebek putih disebelah kanan
- ☐ Yang lain:

< Kembali

Lanjutan >

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

* Wajib

Selanjutnya

Citra 04



Pada citra 04, manakah objek yang paling menarik perhatian menurut anda *

- ☐ 3 sapi hitam didepan
- ☐ sapi putih sendirian di sebelah kanan
- ☐ rumah
- ☐ tanaman hijau disebelah kiri
- ☐ Yang lain:

« Kembali

Lanjutkan »

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

*Wajib

Selanjutnya

Citra 05



Pada citra 05, manakah objek yang paling menarik perhatian menurut anda *

- ☐ wajah singa tengah yang kelihatan taringnya
- ☐ 2 singa disebelah kanan
- ☐ wajah singa yang menganga ditengah
- ☐ 4 singa
- ☐ Yang lain:

« Kembali

Lanjutkan »

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

*Wajib

Selanjutnya

Citra 06



Pada citra 06, manakah objek yang paling menarik perhatian menurut anda *

- ☐ jerapah yang melihat ke arah pembaca
- ☐ 3 jerapah yang tinggi
- ☐ jerapah yang kecil di pojok kanan
- ☐ 4 jerapah
- ☐ Yang lain:

« Kembali

Lanjutkan »

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

*Wajib

Selanjutnya

Citra 07



Pada citra 07, manakah objek yang paling menarik perhatian menurut anda *

- ☐ 3 wajah jerapah
- ☐ jerapah yang disebelah kiri
- ☐ jerapah yang disebelah kanan
- ☐ Yang lain:

◀ Kembali

Lanjutkan ▶

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

* Wajib

Selanjutnya

Pada citra 08, manakah objek yang paling menarik perhatian menurut anda *

- ☐ batu tinggi besar disebelah kanan
- ☐ 3 payung diatas perahu
- ☐ penumpang dibelakang perahu
- ☐ batu disebelah kiri
- ☐ Yang lain:

Citra 08



Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

* Wajib

Selanjutnya

Citra 09



Pada citra 09, manakah objek yang paling menarik perhatian menurut anda *

- ☐ kepala botak
- ☐ 2 orang penyanyi
- ☐ kumpulan audience dibawah
- ☐ kain putih disebelah kanan atas
- ☐ Yang lain:

« Kembali

Lanjutkan »

Lanjutan Kuesioner Deteksi *Saliency*

Survei Visi Komputer

* Wajib

Selanjutnya

Citra 10



Pada citra 10, manakah objek yang paling menarik perhatian menurut anda ?

- ☐ rumah besar ditengah
- ☐ 2 rumah ditengah
- ☐ kabut diatas rumah
- ☐ gunung dibelakang rumah
- ☐ Yang lain:

« Kembali

Kirim »

Hasil Kuesioner Deteksi Saliency

Citra 01



Citra 02

Pada citra 02, manakah objek yang paling menarik perhatian menurut anda



Citra 03

Pada citra 03, manakah objek yang paling menarik perhatian menurut anda



Citra 04

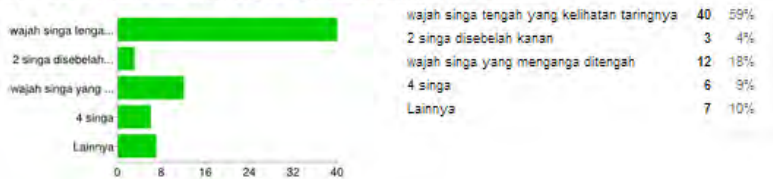
Pada citra 04, manakah objek yang paling menarik perhatian menurut anda



Lanjutan Hasil Kuesioner Deteksi Saliency

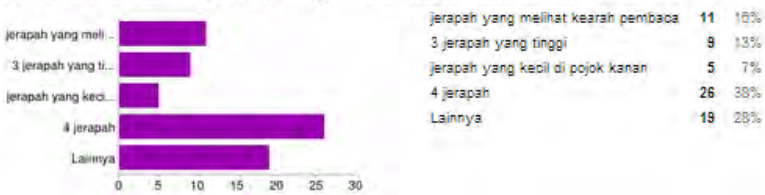
Citra 05

Pada citra 05, manakah objek yang paling menarik perhatian menurut anda



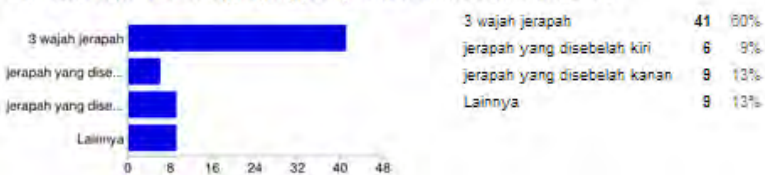
Citra 06

Pada citra 06, manakah objek yang paling menarik perhatian menurut anda



Citra 07

Pada citra 07, manakah objek yang paling menarik perhatian menurut anda



Citra 08

Pada citra 08, manakah objek yang paling menarik perhatian menurut anda



Lanjutan Hasil Kuesioner Deteksi Saliency

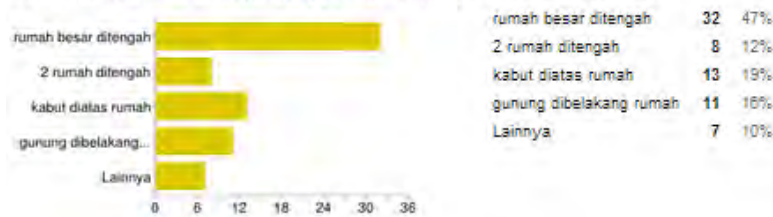
Citra 09

Pada citra 09, manakah objek yang paling menarik perhatian menurut anda



Citra 10

Pada citra 10, manakah objek yang paling menarik perhatian menurut anda



Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Visual *Saliency*, Scholarpedia
<URL:http://www.scholarpedia.org/article/Visual_salience>.
Diakses 10 Juni 2014.
- [2] Hou, X., Zhang, L. “*Saliency Detection: A Spectral residual Approach*”, CVPR, 2007.
- [3] H. Barlow. Possible Principles Underlying the Transformation of Sensory Messages. *Sensory Communication*, pages,217–234, 1961.
- [4] A. Torralba and A. Oliva. Depth Estimation from Image Structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,24(9):1226–1238, 2002.
- [5] HA. Torralba and A. Oliva. Statistics of Natural Image Categories. *Network: Computation in Neural Systems*,14(3):391–412, 2003.
- [6] A. Torralba. Modeling Global Scene Factors in Attention. *Journal of the Optical Society of America*, 20(7):1407–1418, 2003.
- [7] A. Oliva and A.Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
- [8] C. Koch and T. Poggio. Predicting the Visual World: Silence is Golden. *Nature Neuroscience*, 2(1):9–10, 1999.
- [9] L. Itti and C. Koch. Computational Modelling of Visual Attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.
- [10] R. Achanta, F. J. Estrada, P. Wils, and S. S`usstrunk. Salient region detection and segmentation. In *ICVS*, pages 66–75, 2008.

BIOGRAFI PENULIS



Arif Khumaidi lahir di Gresik pada tanggal 20 Mei 1989, merupakan anak ketiga dari empat bersaudara. Pendidikan pertama, bersekolah di TK Al Asyhar Bungah Gresik. Kemudian dilanjutkan jenjang pendidikan sekolah dasar yang berada di MI Al Asyhar Bungah Gresik. Setelah itu dilanjutkan ke MTs Al Asyhar Bungah Gresik dan kemudian di SMA Assa'adah Bungah Gresik. Saat berada di SMA penulis aktif di beberapa kompetisi matematika dan farmasi dan berhasil meraih juara 1 pada bidang matematika di UNISMA Malang tahun 2007. Setelah lulus dari SMA, pendidikan dilanjutkan ke Institut Teknologi Sepuluh Nopember di Jurusan Teknik Elektro dan memilih bidang studi Teknik Komputer dan Telematika. Penulis menjadi anggota aktif SSCP Associate sejak 17 Februari 2014.